PR #35060 完整报告

sgl-project/sglang

Clean up environ.py: remove dead env vars, unify deprecation handling, move examples to a unit test

合并时间：2026-08-17 21:53

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35060

执行摘要

- 一句话：清理 environ.py：统一弃用注册表、删死变量并修复布尔解析
- 推荐动作：值得精读。三个看点：一是声明式弃用注册表的建模方式，用 replacement + transform + note 三元描述覆盖重命名、极性反转、单位换算、直接移除、CLI 迁移提示五类场景，比散落的过程式 warnings 更易审计；二是 review 中 " 删干净而非保留兼容层 " 的取舍，直接删除 SGLANG_NPU_FUSED_MOE_MODE 全部兼容代码是值得借鉴的维护哲学；三是清理类 PR 同样携带行为测试（极性反转、"0" 解析修复），说明重构应与行为验证同步落地。

功能与动机

PR body 明确指出 environ.py 长期积累四类杂乱：无人读取的废弃描述符、四种并存的废弃变量告警机制、以及模块底部约 80 行仅能靠 python environ.py 手工运行的示例函数。这些既是维护负担，也让新增弃用没有统一入口；同时存量映射存在 ' 复制值但不反转极性 ' 的隐性 bug，需要借清理一并修正。

实现拆解

1. 统一弃用处理机制：在 python/sglang/srt/environ.py 引入 _DeprecatedEnv 类与 _DEPRECATED_ENVS 注册表，取代 _print_deprecated_env、_warn_deprecated_env_to_cli_flag、内联 ms→s 循环和 SGL_ 前缀重写四种机制。注册表按 replacement（新变量名）、transform（值变换函数）、note（附加说明）三要素声明式描述每一条弃用项，_handle_deprecated_envs() 在 import 时一次性处理并统一警告格式。
2. 删除死亡描述符与过期 pass：移除 SGLANG_CUTLASS_MOE（连带删除 arg_groups/overrides.py 的 _cutlass_moe_env_override pass、server_args.py 中的调用以及 test_model_overrides.py 的对应测试）、SGLANG_OPT_USE_COMPRESSOR_V2、SGLANG_DISABLE_TP_MEMORY_INBALANCE_CHECK（后者在注册表中保留旧名并做极性反转兼容）。SGLANG_DEEPEP_BF16_DISPATCH 与 DEEP_NORMAL_MODE_USE_INT8_QUANT 因仍有读取方而保留，仅澄清注释。
3. 移除 SGLANG_NPU_FUSED_MOE_MODE 兼容层：删除 server_args.py 的 _has_cli_arg 与 _apply_fuseep_mode_env_compat（以及 resolution pipeline 里的 _handle_fuseep_mode_env_compat），fuseep_mode 恢复为纯 CLI 参数、默认值 2；4

个 NPU 测试文件由环境变量改用显式 --fuseep-mode，docs/hardware-platforms/ascend-npus/model-deployment/best-practices/minimax_m2_5.mdx 同步更新。

4. 修复导入卫生与布尔解析：disaggregation/common/staging_buffer.py 从裸 os.environ.get 改走 envs.SGLANG_STAGING_USE_TORCH.get()，修复 "0" 被 truthy 解析为启用 torch fallback 的问题；debug_utils.cuda_coredump 改为按 SGLANG_CUDA_COREDUMP flag 条件导入，保持 environ 默认导入路径 stdlib-only、无包副作用。
5. 测试与文档迁移：新增 test/registered/unit/test_environ.py（覆盖 EnvField 的 set/get/clear/is_set 语义、override 恢复、subprocess 继承、隐式 bool 报错，以及注册表的移除 / 重命名 / 极性反转 / 单位换算行为），注册进 base-a-test-cpu CI 套件；删除 environ.py 底部 examples() 系列；docs/docs/references/environment_variables.mdx 移除已废弃变量行。

关键文件：

- python/sglang/srt/environ.py（模块 环境变量；类别 source；类型 core-logic；符号 _print_deprecated_env, _DeprecatedEnv, _warn_deprecated_env_to_cli_flag, init）：核心变更文件：引入 _DeprecatedEnv 与 _DEPRECATED_ENVS 声明式注册表，删除 SGLANG_CUTLASS_MOE / SGLANG_OPT_USE_COMPRESSOR_V2 / SGLANG_DISABLE_TP_MEMORY_INBALANCE_CHECK 三个死亡描述符，新增 _invert_bool / _ms_to_s 值变换，并由 _handle_deprecated_envs 在 import 时统一处理。
- test/registered/unit/test_environ.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestEnvField, setUp, test_set_get_clear_is_set, test_set_to_none_is_distinct_from_clear）：新增测试文件：将 environ.py 底部约 80 行手写示例迁移为 EnvField 语义测试，并新增注册表的移除 / 重命名 / 极性反转 / 单位换算测试，注册进 base-a-test-cpu CI，是本 PR 唯一的行为验证入口。
- python/sglang/srt/server_args.py（模块 启动参数；类别 source；类型 core-logic；符号 _has_cli_arg, _apply_fuseep_mode_env_compat）：删除 _has_cli_arg、_apply_fuseep_mode_env_compat 及 _handle_fuseep_mode_env_compat（resolution pipeline 内），并移除 _cutlass_moe_env_override 的 pass 调用，fuseep_mode 恢复为纯 CLI 参数。
- python/sglang/srt/arg_groups/overrides.py（模块 参数覆盖；类别 source；类型 core-logic；符号 _cutlass_moe_env_override）：删除 _cutlass_moe_env_override pass：SGLANG_CUTLASS_MOE 的弃用处理收口到 environ.py 的注册表，不再在参数覆盖管线中静默切换 backend。
- python/sglang/srt/disaggregation/common/staging_buffer.py（模块 暂存缓冲；类别 source；类型 dependency-wiring）：SGLANG_STAGING_USE_TORCH 从裸 os.environ 改走 envs 统一解析，修复字符串 "0" 被 truthy 判定为启用 torch fallback 的行为 bug。
- test/registered/unit/test_model_overrides.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_cutlass_moe_env_override_pass）：删除 test_cutlass_moe_env_override_pass：SGLANG_CUTLASS_MOE 的 backend 切换逻辑已整体移除，原测试目标由 test_environ.py 的注册表告警测试承接。

- docs/docs/references/environment_variables.mdx（模块 文档；类别 other；类型 configuration）：同步删除 SGLANG_CUTLASS_MOE 与 SGLANG_ENABLE_GRPC 两行过期环境变量引用，避免文档与实现脱节。
- test/registered/npu/accuracy/minimax_m2_5/test_npu_minimax_m2_5_w8a8_4p_in64k_out1k_prefix90_50ms_gpqa.py（模块 NPU 测试；类别 test；类型 test-coverage）：代表 4 个 NPU 测试文件（minimax_m2_5 两个、qwen3_235b、speculative_moe_a2a_backend）：SGLANG_NPU_FUSED_MOE_MODE 环境变量改为显式 --fuseep-mode CLI 参数传值。

关键符号：_DeprecatedEnv.apply, _handle_deprecated_envs, _invert_bool, _ms_to_s, _cutlass_moe_env_override, _apply_fuseep_mode_env_compat, prepare_server_args

关键源码片段

python/sglang/srt/environ.py

核心变更文件：引入 _DeprecatedEnv 与 _DEPRECATED_ENVS 声明式注册表，删除 SGLANG_CUTLASS_MOE / SGLANG_OPT_USE_COMPRESSOR_V2 / SGLANG_DISABLE_TP_MEMORY_INBALANCE_CHECK 三个死亡描述符，新增 _invert_bool / _ms_to_s 值变换，并由 _handle_deprecated_envs 在 import 时统一处理。

```
# _DeprecatedEnv: 每个已弃用环境变量的声明式描述。
# - replacement: 新变量名；为 None 表示直接移除无替代。
# - transform: 转发前对旧值的可选变换（极性反转、单位换算等）。
# - note: 附加到警告消息的说明文字。
class _DeprecatedEnv:
    def __init__(
        self,
        replacement: Optional[str] = None,
        transform: Optional[Callable[[str], str]] = None,
        note: Optional[str] = None,
    ):
        self.replacement = replacement
        self.transform = transform
        self.note = note

    def apply(self, old_name: str):
        if old_name not in os.environ:
            return
        # 所有弃用共用一条警告模板，保证输出格式可预期
        message = f"Environment variable {old_name} is deprecated."
        if self.replacement is not None:
            message += f" Please use {self.replacement} instead."
        if self.note is not None:
            message += f" {self.note}"
        warnings.warn(message)
        if self.replacement is not None:
            value = os.environ[old_name]
            if self.transform is not None:
```

```
                value = self.transform(value)
            os.environ[self.replacement] = value # 值转发到新变量


# 单位换算：毫秒 -> 秒（旧 SGLANG_QUEUED_TIMEOUT_MS 系列）
def _ms_to_s(value: str) -> str:
    return str(float(value) / 1000.0)


# 极性反转：DISABLE 类旧变量映射到 ENABLE 类新变量时，
# 旧值 "1" 应转换为新值 "0"，反之亦然
#（旧实现直接复制值导致 SGL_DISABLE...=1 反而启用了检查）
def _invert_bool(value: str) -> str:
    return "0" if value.lower() in ("true", "1", "yes", "y") else "1"


# 唯一的弃用注册表，在 import 时统一处理；
# 新增弃用一律在此登记，禁止散落 ad-hoc warnings
_DEPRECATED_ENVS: Dict[str, _DeprecatedEnv] = {
    "SGLANG_GC_LOG": _DeprecatedEnv(replacement="SGLANG_LOG_GC"),
    "SGLANG_CUTEDSL_MOE_NVFP4_DISPATCH": _DeprecatedEnv(
        replacement="SGLANG_MOE_NVFP4_DISPATCH"
    ),
    # DISABLE -> ENABLE 极性相反，必须经 _invert_bool 转换
    "SGLANG_DISABLE_TP_MEMORY_INBALANCE_CHECK": _DeprecatedEnv(
        replacement="SGLANG_ENABLE_TP_MEMORY_INBALANCE_CHECK",
        transform=_invert_bool,
    ),
    # 重命名且单位变化：毫秒 -> 秒
    "SGLANG_QUEUED_TIMEOUT_MS": _DeprecatedEnv(
        replacement="SGLANG_REQ_WAITING_TIMEOUT",
        transform=_ms_to_s,
        note="Note the unit change: milliseconds -> seconds.",
    ),
    # 直接移除，无替代变量
    "SGLANG_OPT_USE_COMPRESSOR_V2": _DeprecatedEnv(),
    # 由 CLI flag 取代，仅提示迁移方向
    "SGLANG_CUTLASS_MOE": _DeprecatedEnv(
        note="Please use '--moe-runner-backend=cutlass' and/or "
        "'--speculative-moe-runner-backend=cutlass' instead."
    ),
    "SGLANG_ENABLE_GRPC": _DeprecatedEnv(
        note="Please use '--grpc-port' to enable the native gRPC server."
    ),
}


def _handle_deprecated_envs():
    # import 时执行一次：先处理注册表中的显式弃用项，
```

```
    # 再把遗留的 SGL_ 前缀重写为 SGLANG_
    for old_name, deprecation in _DEPRECATED_ENVS.items():
        deprecation.apply(old_name)

    for key, value in list(os.environ.items()):
        if key.startswith("SGL_") and key not in _DEPRECATED_ENVS:
            new_key = key.replace("SGL_", "SGLANG_", 1)
            warnings.warn(
                f"Environment variable {key} is deprecated, please use {new_key}"
            )
            os.environ[new_key] = value
```

test/registered/unit/test_environ.py

新增测试文件：将 environ.py 底部约 80 行手写示例迁移为 EnvField 语义测试，并新增注册表的移除 / 重命名 / 极性反转 / 单位换算测试，注册进 base-a-test-cpu CI，是本 PR 唯一的行为验证入口。

```
# 极性反转回归测试：旧 DISABLE 变量 = 1 时必须让 ENABLE 变量变为 False
# （旧实现直接复制值，SGL_DISABLE...=1 反而启用了内存不均衡检查）
def test_disable_tp_imbalance_check_polarity_is_inverted(self):
    old_name = "SGLANG_DISABLE_TP_MEMORY_INBALANCE_CHECK"
    new_name = "SGLANG_ENABLE_TP_MEMORY_INBALANCE_CHECK"
    os.environ[old_name] = "1"
    self.addCleanup(os.environ.pop, old_name, None)
    self.addCleanup(os.environ.pop, new_name, None)

    self._apply(old_name, _DEPRECATED_ENVS[old_name])
    self.assertIs(envs.SGLANG_ENABLE_TP_MEMORY_INBALANCE_CHECK.get(), False)


# 单位换算测试：毫秒值 1500 经 registry 转发后应变为 1.5 秒
def test_ms_to_s_transform(self):
    old_name = "SGLANG_QUEUED_TIMEOUT_MS"
    os.environ[old_name] = "1500"
    self.addCleanup(os.environ.pop, old_name, None)
    self.addCleanup(os.environ.pop, "SGLANG_REQ_WAITING_TIMEOUT", None)

    self._apply(old_name, _DEPRECATED_ENVS[old_name])
    self.assertEqual(envs.SGLANG_REQ_WAITING_TIMEOUT.get(), 1.5)
```

评论区精华

review 中 merrymercy 自己担任审核者，核心交锋集中在两处：

> "let us just delete these SGLANG_NPU_FUSED_MOE_MODE related code. no need to keep any code for compatibility checks. remove all useless code and keep them clean. just need to take care of the latest usage and keep it clean."

作者明确反对为兼容层保留任何代码，要求直接删除而非维护兼容检查；后续 commit 1b8b00e975 完整落实，fuseep_mode 回归干净的 CLI-only 语义。

另一条针对 environ.py 的评论是简短的 "drop these lines"，作者理解为删除显式的 SGL_DISABLE_TP_MEMORY_INBALANCE_CHECK 注册条目，并回复 "Resolved in 1b8b00e975: dropped the explicit ... registry entry as requested. The polarity test now covers the canonical deprecated SGLANG_DISABLE_TP_MEMORY_INBALANCE_CHECK entry."——即用规范弃用名做极性测试，而不是为旧前缀另立条目。

- SGLANG_NPU_FUSED_MOE_MODE 兼容层是保留还是删除 (design): 在 commit 1b8b00e975 整体移除兼容层、校验与警告，fuseep_mode 恢复为 CLI-only 参数（默认值 2），NPU 测试改为显式传 --fuseep-mode。
- environ.py 中冗余注册条目的删除 (design): 已删除 SGL_ 旧名条目，保留规范弃用名的反转测试，测试与实现命名对齐。
- 废弃环境变量行为变更的测试覆盖 (testing): test_environ.py 新增 test_disable_tp_imbalance_check_polarity_is_inverted 与 test_ms_to_s_transform。

风险与影响

- 风险：
  1. 环境变量行为变更：SGLANG_DISABLE_TP_MEMORY_INBALANCE_CHECK 旧写法的极性被修正，此前设置为 1 反而会启用检查，修复后真正禁用；依赖旧语义（即使是被 bug 驱动的语义）的部署会观测到行为变化。
  2. 弃用变量静默失效：SGLANG_CUTLASS_MOE 从 " 静默切换 MoE backend" 变为 " 仅告警、不再生效 "，存量用户若未迁移到 --moe-runner-backend=cutlass 会看到行为回退；SGLANG_NPU_FUSED_MOE_MODE 被完全移除，NPU 用户如不显式传 --fuseep-mode 将固定走默认值 2。
  3. import 时机风险：_handle_deprecated_envs() 在模块 import 时写 os.environ 并触发 warnings，而 environ.py 是全库最早加载的模块之一；任何以 SGL_ 开头的第三方环境变量都会被重写（该逻辑此前已存在，本次只是统一收口，风险等级未上升但值得留意）。
  4. CI 信号待确认：PR body 显示 Extra 测试为 :x:，与 4 个 NPU 测试改为 --fuseep-mode 的配套改动相关，NPU 侧回归需在真实硬件上确认。 - 影响：影响范围：environ.py 被整个 SGLang 进程树早期导入，新弃用注册表会影响所有启动路径；警告消息格式统一为 "Environment variable X is deprecated..."，便于用户识别迁移方向。用户影响：使用被弃用变量的存量用户会在启动时看到新格式警告，其中 SGLANG_CUTLASS_MOE 与 SGLANG_NPU_FUSED_MOE_MODE 两类用户必须迁移到 CLI flag，否则行为不再生效。团队影响：新增弃用变量从此只需在 _DEPRECATED_ENVS 加一行声明，禁止再散落 ad-hoc warnings，显著降低维护成本；同时 environ.py 从 " 带示例的模块 " 变成 " 纯机制模块 "，测试入口收敛到 test_environ.py。 - 风险标记：核心基础设施变更，环境变量行为变更，NPU 配置迁移，CI Extra 未通过

关联脉络

- PR #35105 Revert "[AMD] [GLM5] Fuse shared-expert append into aiter grouped-topk (skip per-layer append kernel)": 弱关联：分支 commit 'Resolve topk revert after main update' 表明合入 main 后需处理 AMD grouped-topk 回滚冲突；虽无文件级功能依赖，但反映主线 kernel 层波动对本次合入的影响。